OPEN ACCESS

University of BRISTOL

Aguado, A., Davis, M. J., Peng, S., Alvarez, M. V., Lopez, V., Szykowiec, T., Autenrieth, A., Vialta, R., Mayoral, A., Munoz, R., Casellas, R., Martinez, R., Yoshikane, N., Tsuritani, T., Nejabati, R., & Simeonidou, D. (2016). Dynamic Virtual Network Reconfiguration Over SDN Orchestrated Multitechnology Optical Transport Domains. *Journal of Lightwave Technology*, *34*(8), 1933-1938. https://doi.org/10.1109/JLT.2016.2522823

Peer reviewed version

Link to published version (if available):
10.1109/JLT.2016.2522823

Link to publication record in Explore Bristol Research
PDF-document

(C) 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/

Dynamic Virtual Network Reconfiguration over SDN Orchestrated Multi-Technology Optical Transport Domains

A. Aguado, M. Davis, S. Peng, M. V. Alvarez, V. Lopez, T. Szyrkowiec, A. Autenrieth, R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, R. Martínez, N. Yoshikane, T. Tsuritani, R. Nejabati and D. Simeonidou

Abstract— **Network virtualization is an emerging technique that enables multiple tenants to share an underlying physical infrastructure, isolating the traffic running over different virtual infrastructures/tenants. This technique aims to improve network utilization, while reducing the complexities in terms of network management for operators. Applied to this context, software defined networking (SDN) paradigm can ease network configurations by enabling network programmability and automation, which reduces the amount of operations required from both service and infrastructure providers. SDN techniques are decreasing vendor lock-in issues due to specific configuration methods or protocols. Application-based Network Operations (ABNO) is a toolbox of key network functional components with the goal of offering application-driven network management. Service provisioning using ABNO may involve direct configuration of data plane elements or delegate it to several control plane modules. We validate the applicability of ABNO to multi-tenant virtual networks in multi-technology optical domains based on two scenarios, in which multiple control plane instances are orchestrated by the architecture. Congestion Detection and Failure Recovery, are chosen to demonstrate fast recalculation and reconfiguration, while hiding the configurations in the physical layer from the upper layer.**

Index Terms—**Optical Packet Switching, Optical Circuit Switching, Software Defined Networking, Network Virtualization, Application-Based Network Operations, Network Monitoring, Reconfiguration.**

I. INTRODUCTION

Optical transport networks are emerging as a key solution to the ever-growing demand for supporting new applications and services (e.g. Ultra-High-Definition Video Streaming), which require high bandwidth and low latency network connectivity. Optical packet switching (OPS) could fit into metro access networks and significantly reduces the number of opto-electro-optic conversions, enabling dynamic switching by using OPS labels. Elastic optical networks (EONs), based on flexi-grid optical circuit switching (OCS), are used to accommodate demands by providing flexible bandwidths over 12.5GHz slices. On the other hand, different sets of network control protocols (such as OpenFlow (OF) and Generalized Multi-Protocol Label Switching (GMPLS)) have been defined to enable/enhance network programmability. The ICT STRAUSS project (Fig. 1) addresses the above technologies by deploying a network virtualization, control and orchestration layers. The proposed OPS technology is analyzed as a possible solution in a Data Center (DC) environment, as well as a possible solution in a metro access network. The OCS fixed/flexi-grid DWDM technology is studied for the transport segment.

The Application-Based Network Operations (ABNO) architecture [1] is a framework that enables network automation and programmability thanks to the utilization of standard protocols and components already defined within the IETF. In our context, the architecture provides software defined network orchestration, meaning network control and resource reservation through several domains and technologies, whilst using standard protocols. The ABNO can sit directly on top of network devices, as well as orchestrate several control plane instances (such as SDN controllers or path computation elements) in order to provide end-to-end connectivity. A first experimental demonstration of the ABNO capabilities to provision end-to-end services over GMPLS and OF enabled multi-layer scenarios using multiple Path Computation Elements (PCE) was done in [2]. A Multi-domain Network Hypervisor (MNH) was developed in [3]. The MNH runs on top of the ABNO architecture. The MNH provides an abstracted and virtual view of the tenant's virtual infrastructure exposing topological information through OF version 1.0 and requests E2E connections from the ABNO. The MNH acts as an interface for network virtualization and control to the infrastructure and service provider. The ABNO is responsible to monitor the assigned QoS to the requested E2E connections, in order to satisfy the Virtual Network (VN) requirements. If a certain degree of QoS cannot be granted to the deployed VN, the ABNO will be responsible for triggering the E2E recovery mechanism. Although the proposed scenario is based on OPS and OCS technologies, the presented architecture can be generalized

Manuscript submitted October 23, 2015. This work was supported by the European Community's Seventh Framework Programme FP7/2007-2013 under Grant n° 608528 and by the Japanese Ministry of Internal Affairs and Communications and the National Institute of Information and Communications Technology through the EU-Japan coordinated project STRAUSS, and the Spanish Ministry of Economy and Competitiveness through the project FARO (TEC2012-38119).

A. Aguado, M. Davis, S. Peng, R. Nejabati and D. Simeonidou are with the High Performance Networks group, Electrical and Electronic Engineering, University of Bristol, U.K. (email: {a.aguado: matt.davis: shuping.peng: reza.nejabati: dimitra.simeonidou}@bristol.ac.uk)

M. V. Alvarez and V. Lopez are with Telefónica I+D c/ Don Ramón de la Cruz, 82-84, 28006 Spain (email: victor.lopezalvarez@telefonica.com).

T. Szyrkowiec and A. Autenrieth are with ADVA Optical Networking 82152 Martinsried, Germany (e-mail: {tszyrkowiec: aautenrieth} @advaoptical.com).

R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas and R. Martínez are with the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), 08860 Barcelona, Spain (email: {ricard.vilalta: arturo.mayoral, raul.munoz: ramon.casellas: ricardo.martinez}@cttc.es).

N. Yoshikane and T. Tsuritani are with KDDI R&D Laboratories, Saitama, 356-8502 Japan (e-mail: {yoshikane: tsuri}@kddilabs.jp).

for multi-layer and/or multi-domain scenarios.

This paper extends the work in [4], where the experimental assessment of the capabilities of the ABNO architecture is presented for two different use cases: (i) Failure recovery and (ii) Congestion detection. For the first time, both are applied over multi-technology optical networks. We demonstrate the replanning of network services that are requested by the MNH, thus making the virtualization layer unaware of any network reconfiguration. These results show the capabilities of the ABNO architecture, which is able to recalculate and reconfigure E2E connections based on the detected network events such as link failure and congestion. Further these results can be applied to create new E2E backup paths when an existing service is moved to a backup path, and a new backup path needs to be created to keep service protection.

This work is organized as follows: Section II focuses on multi-domain network virtualization. The control plane layers involved (orchestration and network control) and the ABNO architecture, as well as the interfaces between components are introduced in section III. Section IV describes an overview of the two use case scenarios. Section V further explains the event handling and corresponding workflows for the two use cases. Finally, section VI concludes our work.

II. MULTI-DOMAIN NETWORK HYPERVISOR

The virtualization of IT resources (CPU, storage) caused a revolution on the IT industry with the emergence of cloud computing [5]. Network virtualization has attracted a lot of attention in the last years [6], aiming to improve resource utilization by partitioning and abstracting the network infrastructure to be offered to multiple tenants as VNs. VN provisioning is a challenging task when heterogeneous networks, which do not naturally interoperate, are involved. A VN might be employed for DC interconnection, and thus likely, it will be deployed across multiple transport technologies (OPS, OCS) with different control planes (OF, GMPLS), which need to be coordinated to offer a uniform transport service.

The authors in [7] have proposed virtual optical network services across multiple domains, but the work did not take into account the inherent heterogeneousness of multiple control domains. In [8], the authors propose a multi-domain resource broker which takes into account this heterogeneity. It also presents an orchestration mechanism, which allows the composition of end-to-end virtual transport infrastructures across different transport technologies as well as end-to-end network service provisioning across multiple virtual optical networks comprising different transport and control plane technologies. The proposed multi-domain resource broker provides per-domain virtualization. It can be observed how each domain is virtualized with a per-domain network hypervisor. A network orchestrator is introduced later in order to provide end-to-end services over the virtualized domains of a single multi-domain virtual optical networks.

Another possible approach, proposed by the authors in [9] and used in this paper, is to provide end-to-end virtualization. The MNH acts as a network hypervisor to deploy multi-tenant VONs and allow their own individual customized control. The MNH runs over a network orchestrator, as shown in Figure 3. The network orchestrator is responsible for providing end-to-end connectivity on top of multi-technology multi-vendor multi-domain transport networks. Considering this approach, the MNH is in-line with the ACTN framework (IETF) [10] and the SDN architecture (ONF) [11].

III. ARCHITECTURE AND INTERFACES

The control plane architecture designed and developed in the EU FP7 project STRAUSS [12] has been presented in [13] (see Fig. 1). It consists of a hierarchical architecture, with multiple functional blocks well-defined and logically-spread within the components, providing E2E service management, as well as infrastructure virtualization over multi-technology optical domains.

The lower layer in the STRAUSS overall architecture is formed by different SDN controllers (such as POX [14] or ODL [15]) controlling OF (version 1.0 with extensions) enabled optical networks, and active and stateful PCE (for GMPLS enabled domains).

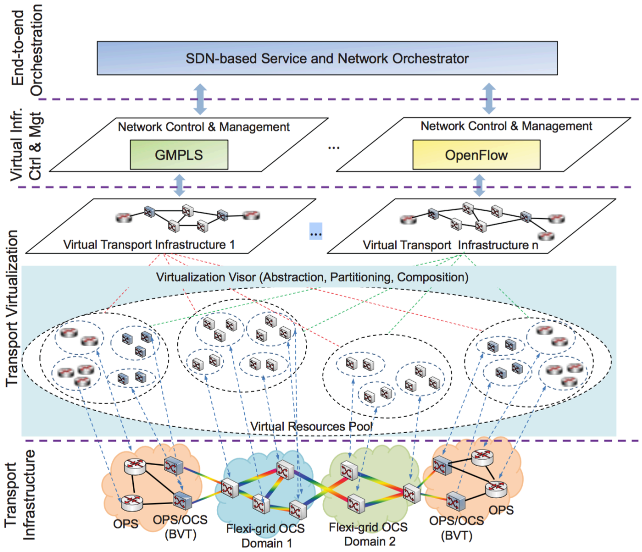

Fig. 1 STRAUSS Control plane architecture

The ABNO architecture sits in the middle of the STRAUSS control plane, acting as a network orchestrator. The architecture is composed of a set of well-defined modules that encapsulate different network functionalities. They utilize standard protocols and interfaces for both internal (PCEP, RESTful) and external communications (RESTful NBI and SBI, GMPLS protocols such as PCEP and BGP-LS). The main components used in this work, shown in Fig. 2, are:

- The *ABNO controller* is the main gateway to the architecture. It is able to receive requests through a REST interface and orchestrate other components in the architecture to map the incoming request to a specific workflow.
- The *PCE* is responsible for the path calculation through the network, based on the current network status and the requirements of the request.
- The *Virtual Network Topology Manager* (VNTM) is the module that stores the multilayer information. It sets up or tears down lower-layer Label Switched Paths (LSPs) as well as upper-layer virtual links consequently.
- The *Topology Module* has at least one topology database with information about the available

network elements such as node types and addresses, controller types and their addresses, edges, ports information, etc. It can gather information through different protocols and interfaces (BGP-LS, OSPF-TE and/or REST APIs from SDN controllers).

- The *Provisioning Manager* is the module that handles the E2E path by configuring the controllers. It splits the explicit route object (ERO) according to the different domains, sending the configuration using different protocols, i.e. OpenFlow, RESTful and/or PCEP Initiate [14] for GMPLS domains.
- The Operations, Administration and Maintenance Handler (OAM-Handler) is responsible for retrieving network monitoring information and handle network events.

A multi-domain network hypervisor (MNH) has been developed on top of the orchestration layer, enabling multi-tenancy over heterogeneous network domains, providing visibility of the customer's network through OF 1.0. The main responsibilities of the module are: a) providing multi-domain network connectivity interfacing with the network orchestrator, b) providing virtual visibility of the network using OF to communicate with customers' SDN controllers and c) mapping virtual and physical infrastructure, translating customer FLOW_MOD messages in connection requests in the physical infrastructure through the network orchestrator.

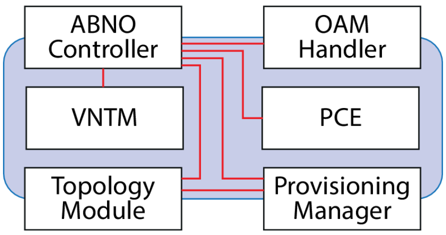

Fig. 2. ABNO Architecture (Modules used for this demonstration).

IV. Use Cases Context

A. *Failure Recovery*

Current services require fast recovery from network failures (<50 ms) [17], this is possible in optical network using protection at the optical level. The ABNO architecture is proposed as a solution for orchestrating networks that have multiple controllers and is able to handle different network events. In this context, the ABNO architecture is able to receive failure alarms and start recovery workflows to create new E2E paths. A distributed control plane usually has configuration time penalties (in our case, around 100ms delay VPN connections, full configuration between 2 and 4 seconds). The provisioning time of this connection cannot meet the 50 ms requirement, but it is enough for best effort services. Moreover, ABNO can create and provision a new backup connection after the first failure is detected. The Operation, Administration and Maintenance (OAM) handler receives failure alarms, triggers an internal workflow, and allows the ABNO controller to obtain information about the affected services and to configure new E2E connections.

B. *Congestion Detection and Replanning*

Real time monitoring and dynamic network reconfiguration have become increasingly attractive due to the growing bandwidth on demand and the latest standards and protocols that act as enablers to provide more flexibility in the network design. Decoupling control from data plane will decrease the complexity in network nodes by removing routing and complex computations from their architectures, but it requires a highly aware control plane, that is able to retrieve and handle a big amount of information, and act in accordance with it. A set of monitoring tools and protocols has been created for detecting congestion in network interfaces. We have also developed a simple tool that acts as an application sitting on top of the SDN controller to retrieve flow information and statistics. It builds a bandwidth usage graph that allows setting adaptive threshold to control the traffic load. In the event that the threshold is exceeded, the application notifies the OAM handler, thereby starting a replanning workflow, moving the affected service to less congested interfaces and/or network domains. To extend the work done in [4], we use monitoring and security focused NIC card based on SolarFlare technologies [18]. This card supports internal software development of modules that are used as a filter chain. For this extension we propose both, reactive (based on websockets subscription) and proactive (via RESTful interface) solutions for the control plane, explaining the advantages of using this technology instead of the one proposed in the previous work.

V. Experimental Setup and Demonstration

The experimental setup is shown in Fig. 3, where Bristol, KDDI and ADVA domains are OF-enabled and controlled by SDN controllers (i.e. ODL, POX), while CTTC is a GMPLS domain managed by an active and stateful PCE. Multi-partner interoperability between control plane instances via GMPLS, including various extensions, has been successfully tested in [19]. For the purpose of this test, the interface between the active PCE and the ABNO is implemented using REST. ABNO GMPLS support can be checked in [20]. The controllers hide the internal setup of each domain from the ABNO [4], allowing an abstract view of the experimental setup (each domain is abstracted as a single node). The abstract topology consists of two OPS domains (KDDI and Bristol) and two flexi-grid OCS domains (ADVA and CTTC). At the edge, an SDN-enabled opto-electrical interface [21] was implemented for interconnecting the OPS and flexi-grid OCS domains. Such an interface can be seen as a L2 switch that retrieves statistics from packet counting. The E2E paths through multiple domains are calculated, delegating the internal computation to each network controller.

A. *Use Case 1: Failure Recovery*

When the network is up and running, multiple virtual networks requested by different tenants have already been set up and E2E connections have been established. The SDN controller (e.g. POX at KDDI) monitors the traffic via optical power meters, therefore it is aware of the status of its own domain. When the optical power reaches a pre-determined threshold (dependent on the transmission system), an OpenFlow agent, which is running on top of KDDI OPS nodes and optical monitors, sends an event to the domain SDN controller. As an example, if the monitored optical power is below -20dBm, the agent sends an alarm to the controller. Once an event is detected by the controller, it sends an HTTP POST message to the OAM handler in the ABNO. The alarm is sent in a JavaScript object notation (JSON) message, with the following format:

{"event":"alarm", "id":"PORT_FAILURE", "body":
{"dpid":"00:00:00:00:00:00:FF:01","port":"3"}}

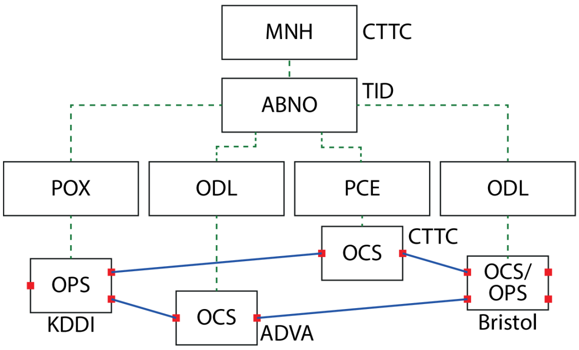

Fig. 3. Experimental Setup.

Where "event" shows the type of message, "id" the reason for it, "dpid" the switch id, and "port" the affected port.
The internal ABNO workflow is given as follows:

1) The OAM handler receives the JSON, parses the message, interprets the event, and maps it to a specific ABNO workflow.

2) The ABNO controller module gets all the affected E2E connections from a service table stored by the architecture.

3) For each affected service, the ABNO controller sends a PCEP request message (see Fig.4a) with a sub-object extension of the exclude route object (XRO) (see Fig. 4b). The XRO contains the information about the failed interface or node, so that the PCE can exclude this resource and calculate the new paths. XRO is chosen to update the PCE TED, which is due to the long response time and synchronization issues for using an REST API or IGP discovery calculating inappropriate paths.

4) The ABNO controller receives the PCEP response and then sends the new computed path to the provisioning manager

5) The provisioning manager splits the Explicit Route Object (ERO) into multiple configuration flows for each domain and control technology (OF/PCEP), creating the new

```
⊟ Path Computation Element communication Protocol
  ⊞ PATH COMPUTATION REQUEST MESSAGE Header
  ⊞ RP object
  ⊞ END-POINT object
  ⊞ BANDWIDTH object
  ⊞ OBJECTIVE FUNCTION object (OF)
  ⊞ EXCLUDE ROUTE object (XRO)
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X| Type = 0x05 |     Length    |           DPID (8 bytes)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       DPID (continued)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        DPID (continued)       |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 4: top) PC Request with XRO, bottom) DPID XRO sub-object

E2E path.
Fig. 5 shows the message flow among the modules inside the ABNO. For instance, referring to Fig. 3, if there are some flows from KDDI to Bristol through CTTC, which fails, all the flows will be rerouted through other domains (i.e. ADVA) and services will be restored. Fig. 6 shows the exchange of messages for both use cases.

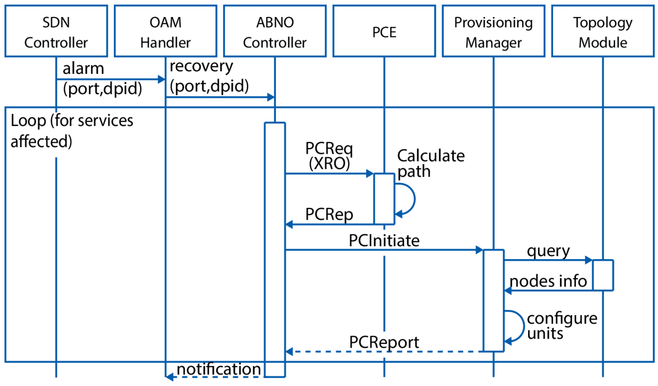

Fig. 5 Failure Recovery Workflow

B. Use Case 2: Congestion Detection and Replanning

We assume an initial network configuration with several E2E connections between the OPS domains. End users start generating traffic using available network services. Our aforementioned packet-counting-based network monitor application sits on top of the SDN controller (e.g. ODL-based in Bristol), which is able to get traffic statistics from the edge interfaces (seen as L2 switches by the controller). It builds a graph with the status of each interface at the edge node. For this purpose, ODL default configuration (flowStatsPollInterval parameter) was modified to force the controller to retrieve flow statistics from the nodes by sending OFPT_STATS_REQUEST messages every second.

10.0.34.106	10.0.34.110	HTTP	POST /oam/v0.0/rest/eventHandler HTTP/1.1
127.0.0.1	127.0.0.1	HTTP	GET /?Source_Node=00:00:00:00:00:00:FF:01&
127.0.0.1	127.0.0.1	PCEP	Path Computation Request (PCReq)
127.0.0.1	127.0.0.1	PCEP	Path Computation Reply (PCRep)
127.0.0.1	127.0.0.1	PCEP	Path Computation LSP Initiate (PCInitiate)
10.0.34.110	10.0.34.106	HTTP	PUT /controller/nb/v2/flowprogrammer/defau
10.0.34.110	10.0.34.104	HTTP/XML	PUT /controller/nb/v2/flowprogrammer/defau
10.0.34.106	10.0.34.110	HTTP	HTTP/1.1 200 OK
10.0.34.104	10.0.34.110	HTTP	HTTP/1.1 200 OK (text/plain)
10.0.34.110	10.0.34.112	HTTP	POST /OF/ HTTP/1.1 (application/json)
10.0.34.112	10.0.34.110	HTTP	HTTP/1.1 200 OK (application/json)
127.0.0.1	127.0.0.1	PCEP	Path Computation LSP State Report (PCRpt)
127.0.0.1	127.0.0.1	HTTP	HTTP/1.1 200 OK (text/html)

Fig. 6 Failure recovery (one single service) and congestion replanning capture

Once an interface exceeds the predefined load threshold (configured by the domain infrastructure provider), the application sends a JSON string. The message specifies the overloaded interface that should be avoided by the PCE in the computation of the replanned path and the flow information to detect the affected E2E service. The JSON format is:

*{"event":"alarm", "id":"CONGESTION", "body":
{"dpid":"00:00:00:00:00:00:30:04","port":"3","flow":{"dp
id":"00:00:00:00:00:00:30:04","ingressPort":"1",s
"output":"3"}}*

Where "event", "id" and "body" are equally structured as in the previous use case, and flow contains information about the static flow affected.

As explained in the previous section, we have included - as an extension of this work - inline monitoring based on a programmable SolarFlare 10G NIC card. This technology allows the infrastructure provider to design the functionalities of the card by implementing several modules that can be concatenated creating a filter chain. This enables the creation of dynamic network functions (such as filters, firewalls, deep packet inspectors, security agents, etc.) that could be assigned to the tenant's virtual network as a virtual network functions. For the purposes of this test, we have implemented a packet counting module that inspects traffic and identifies frame by IP. This module keeps track of the traffic generated by two virtual machines, building a data rate graph for both. It allows the user to access the monitored data in two different ways: a proactive way, by accessing the data using a RESTful interface (HTTP GET message); and a reactive way, by subscribing to threshold-exceeded events using a websocket. Both interfaces are located in a server that hosts the NIC card.
Using this additional solution we improve the monitoring function by solving two issues detected on the initial solution: detection time (that depends on timers configured in the controller and the application) and mismatches among agent, controller and application. This second issue is generated by different timers configured in the application and ODL, and process time overhead in both, producing eventually wrong data (mismatched data). This can be reduced by increasing the application-to-controller polling time, which increases the first listed problem (response time).
Subscription to the NIC card can improve the reaction time up to N+M seconds, where N is the polling time from the controller to the devices/agents and M is the polling time from the application to the controller. The proposed subscription and polling-based methods avoid at the same time any possible mismatching by using the NIC card.
Figure 7 shows the network setup that consists in a one to three optical splitter (6dB loss for the output to SolarFlare) and the NIC card. Fig. 8 shows the internal workflow among the modules inside the ABNO.

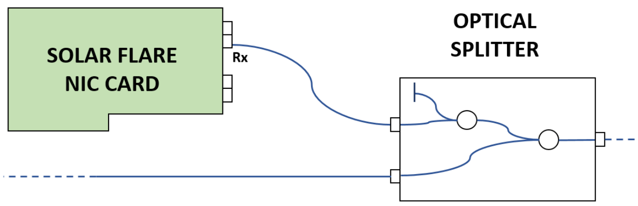

Fig. 7 SolarFlare setup

The internal flow within the orchestrator is as follows:
1) The OAM handler receives the alarm and sends the event to the ABNO controller, starting the appropriate workflow.
2) The ABNO controller starts a replanning workflow and obtains the specific service that matches the flow from the service table.
3) A single PCEP request with the XRO is sent to the PCE, which avoids the overloaded interface.
4) The PCEP response with the E2E path is forwarded from the ABNO controller to the provisioning manager.
5) The provisioning manager finally splits the path according to the boundaries of each traversed domain and then sends the configuration to the responsible controllers.

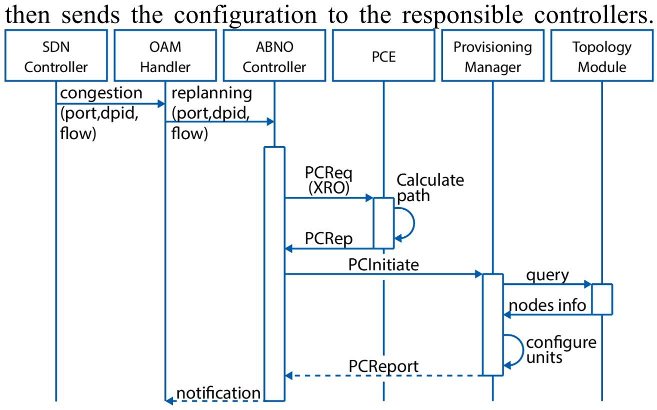

Fig. 8 Congestion detection and replanning workflow

Fig. 9a) shows the statistics that our monitoring application is retrieving (load of the interface number 3 over time), while 9b) shows the traffic through all the deployed flows in the switch, captured with an independent application. Both apps are running as separate processes with the same polling intervals, but not polling at the same time. This causes a one second disparity between the two graphs, because both snapshots were taken at the same instant, when the second application did not retrieve its last value. When the first flow starts generating more traffic, the total load of the interface overpasses the threshold. This event is detected by the application. The application then automatically starts the appropriate workflow through the ABNO using the OAM handler's RESTful server. Figs. 9(a) and (b) show how the flow that generates the highest amount of the traffic (flow from interface 2 to 3 in the Bristol edge node) is moved from one interface to the other available one (number 4) within the OPS domain. These ports are mapped to the four ports from

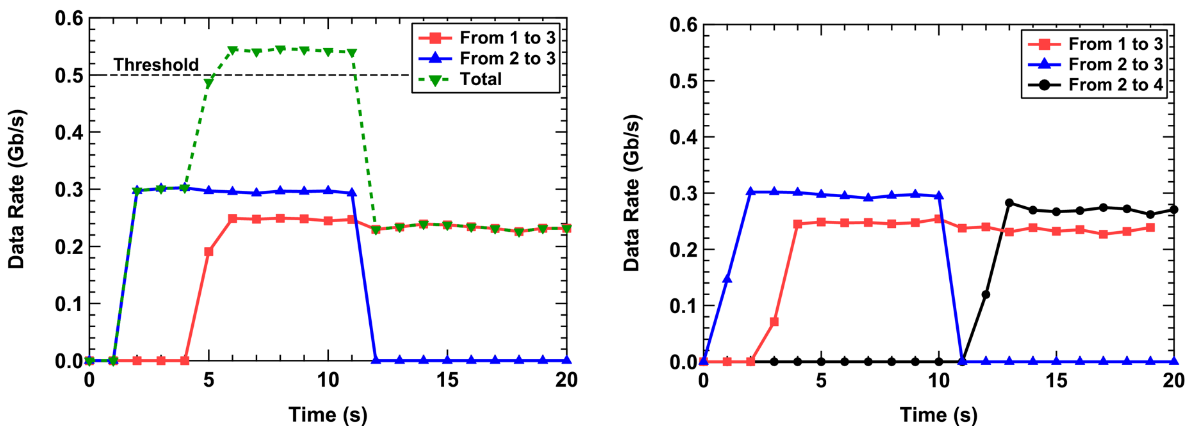

Fig. 9: a) Load of Interface 3 b) Load generated by all flows

the Bristol domain (1-2 acting as ingress interfaces, while interface 3 goes to CTTC domain and 4 to ADVA. See Fig. 3).

VI. CONCLUSIONS

This work presents, for the first time, the applicability of the ABNO architecture to two scenarios: (i) failure recovery and congestion replanning over SDN orchestrated multi-technology optical domains based on optical power and (ii) inline layer 2 monitoring, respectively. We validate the ABNO architecture as a key part of replanning multi-tenant and multi-technology optical network domains by experimentally demonstrating its capabilities in a distributed network. This allows multiple users/tenants to rely on our control plane in order to create, maintain and replan their virtual infrastructures without awareness of any reconfiguration. In addition, we have studied two different methods for the packet counting based monitoring: OpenFlow flow statistics using the ODL NBI and a separate monitoring link using an optical splitter and a NIC card. As such, this work illustrates the major advantages of working with inline layer 2 monitoring based on a NIC card, which provides a substantial improvement in terms of both the reaction time (in seconds, depending on the poll timer of the ODL application) and accuracy (avoiding mismatches that are inversely proportional to the previously mentioned poll timers).

REFERENCES

[1] D. King and A. Farrel “A PCE-Based Architecture for Application-Based Network Operations”, RFC 7491, 2015.
[2] A.Aguado et al., “ABNO: A feasible SDN Approach for Multivendor IP and Optical Networks”, in OFC, Mar 2014.
[3] R.Vilalta et al., “Network Virtualization Controller for Abstraction and Control of OpenFlow-enabled Multi-tenant Multi-technology Transport Networks” OFC 2015.
[4] A. Aguado et al., “Dynamic Virtual Network Reconfiguration over SDN Orchestrated Multi-Technology Optical Transport Domains” in ECOC 2015
[5] A Vouk, M. (2008). Cloud computing–issues, research and implementations. *CIT. Journal of Computing and Information Technology*, *16*(4), 235-246.
[6] Nejabati, R., Escalona, E., Peng, S., & Simeonidou, D. (2011, February). Optical network virtualization. In *Optical Network Design and Modeling (ONDM), 2011 15th International Conference on* (pp. 1-5). IEEE.
[7] Y.Wang, Y. Jin,W. Guo,W. Sun, and W. Hu, “Virtualized optical network services across multiple domains for grid applications,” IEEE Communications Magazine, 2011.
[8] R. Vilalta, R. Muñoz, R. Casellas, R. Martínez, S. Peng, M. Channegowda, T. Vlachogiannis, R. Nejabati, D. E. Simeonidou, X. Cao, T. Tsuritani, and I. Morita, “Dynamic multidomain virtual optical networks deployment with heterogeneous control domains,” IEEE/ OSA Journal of Optical Communications and Networking, vol. 7, no. 1, pp. A135 – A141, 2015.
[9] Ricard Vilalta, Raul Muñoz, Ramon Casellas, Ricardo Martinez, Shuping Peng, Reza Nejabati, Dimitra Simeonidou, Noboru Yoshikane, Takehiro Tsuritani, Itsuro Morita, Victor Lopez, Thomas Szyrkowiec, Achim Autenrieth, Multidomain Network Hypervisor for Abstraction and Control of OpenFlow-Enabled Multitenant Multitechnology Transport Networks [Invited], Journal of Optical Communications and Networking, vol. 7, issue 11, November 2015.
[10] Y. Lee, D. Dhody, S. Belotti, K. Pithewan, and D. Ceccarelli, “Requirements for abstraction and control of transport networks,” in draft-lee-teas-actn-requirements-00.txt. IETF, 2015.
[11] ONF, “Sdn architecture issue 1,” ONF, Tech. Rep., 2014.
[12] EU FP7 project STRAUSS. Available: http://www.ict-strauss.eu
[13] R. Munoz “Network virtualization, control plane and service orchestration of the ICT STRAUSS project” in EuCNC, 2014.
[14] POX SDN controller. Available: http://www.noxrepo.org/pox/about-pox/
[15] OpendayLight controller. Available: https://www.opendaylight.org/
[16] E. Crabbe “PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model”, IETF draft.
[17] B. Niven-Jenkins et al “Requirements of an MPLS Transport Profile”, RFC 5654, 2009
[18] SolarFlare. Available: http://solarflare.com/
[19] O. Gonzalez de Dios, et al. “First Multi-partner Demonstration of BGP-LS enabled Inter-domain EON control with H-PCE” in OFC, Mar 2015.
[20] Github netphony-network-protocols https://github.com/telefonicaid/netphony-network-protocols
[21] Y.Yoshida et al., “First international SDN-based Network Orchestration of Variable-capacity OPS over Programmable Flexi-grid EON” OFC 2014.